On Regular Message Sequence Chart Languages and Relationships to Mazurkiewicz Trace Theory

Rémi Morin*

Institut für Algebra, Technische Universität Dresden, D-01062 Dresden, Germany

Abstract. Hierarchical Message Sequence Charts are a well-established formalism to specify telecommunication protocols. In this model, numerous undecidability results were obtained recently through algebraic approaches or relationships to Mazurkiewicz trace theory. We show how to check whether a rational language of MSCs requires only channels of finite capacity. In that case, we also provide an upper bound for the size of the channels. This enables us to prove our main result: one can decide whether the iteration of a given regular language of MSCs is regular if, and only if, the Star Problem in trace monoids (over some restricted independence alphabets) is decidable too.

Message Sequence Charts (MSCs) are a popular model often used for the documentation of telecommunication protocols. They profit by a standartized visual and textual presentation (ITU-T recommendation Z.120 [20]) and are related to other formalisms such as sequence diagramms of UML [5] or message flow diagramms. An MSC gives a graphical description of the intended communications between processes. It abstracts away from the values of variables and the actual contents of messages. However, this formalism can be used at a very early stage of design to detect errors in the specification [18]. In this direction, several studies have already brought up methods and complexity results for the model checking of MSCs viewed as a specification language [14,26,27,1]. However, many undecidable problems arose by algebraic reductions to formal language theory [7] or relationships to Mazurkiewicz trace theory [28,16].

We are here interested in *regular* sets of MSCs, a notion recently introduced in [16]. These languages of MSCs are such the set of associated sequential executions can be described by a finite automaton; therefore model checking becomes decidable and particular complexity results could be obtained [1,28]. Moreover, regular languages of MSCs satisfy the *channel-bounded property*, that is, the number of messages stored in channels at any stage of any execution is bounded by a finite natural number. Consequently, as shown in [17,25], regular languages of MSCs admit a finite distributed abstract implementation in the form of message passing automata whose channels have a finite capacity; noteworthy, this result relies on asynchronous mappings [8] studied in order to associate a finite cellular asynchronous automaton to any recognizable subset of Mazurkiewicz traces. Another interesting characterization of regular MSC languages was established in [17]: they are precisely the languages that are definable in Monadic

* Supported by PROCOPE.

F. Honsell and M. Miculan (Eds.): FOSSACS 2001, LNCS 2030, pp. 332–346, 2001.

Second Order logic and which satisfy the channel-bounded property. Again, this result relies on technical results from trace theory.

It is the aim of this paper to contribute to this stream of relationships between traces and MSCs. These two formalisms consist of labelled partial orders (also known as pomsets [32]) provided with a concatenation that yields a monoidal structure. A *Hierarchical (or High-level) Message Sequence Chart (HMSC)* is then a description of a set of MSCs built from finite sets by use of union, concatenation (i.e. products), and iteration. Thus an HMSC is simply a rational expression in the monoid of MSCs. Moreover any HMSC can be flattened into a Message Sequence Graph (MSG), which is a kind of finite automaton labelled by MSCs. However, HMSCs are more succint descriptions than MSGs and they are also closer to the hierarchical specification of MSC languages. On the other hand, MSGs are often more convenient for the study of languages of MSCs [1, 28,16]. Since MSGs and HMSCs have the same expressive power, most results known for MSGs apply also to HMSCs. In [28], correction and consistency of a given HMSC are shown to be undecidable; more important here, it is shown in [16] that *one cannot decide whether a given HMSC describes a regular language of MSCs.* Both negative results rely actually on the undecidable Closure Problem in trace monoids [33]. A natural approach is now to restrict the class of HMSCs to be used, so that interesting properties of the associated languages are ensured or can effectively be checked.

In this direction, a particular class of HMSCs, called *locally synchronized* in [28] and *bounded* in [1] restricts iteration to sets of MSCs whose communication graphs are strongly connected. These sc-HMSC proved to be interesting since the associated languages are regular [28,1]. The converse was shown in [16]: any regular finitely generated language of MSCs can be described by an sc-HMSC. We shall see here how both relationships can be infered from Ochmański's theorem [31]. In our way, we show how one should adapt the definition of communication graph in order to deal with internal actions.

Our first result asserts that an HMSC satisfies the channel-bounded property if, and only if, iteration occurs only over sets of MSCs for which each connected component of the communication graph is strongly connected. Therefore *divergence of channels is easily decidable.* This evokes Ben-Abdallah & Leue static criterion [4] to check divergence freeness of HMSCs, although divergence freeness and channel-boundedness are in general distinct notions. Our proof also differs from [3,4] by providing a technically usefull upper bound for the size of the channels.

As mentionned above, regularity is however undecidable [16]. We consider in this paper a variation of this problem: we would like to check whether each subexpression of a given HMSC describes a regular language — and not only the whole HMSC itself. Since unions and products of regular languages are regular, we consider here the following problem for MSCs: *given a regular language of MSCs, decide whether its iteration is regular too.* Our second result asserts that *this problem is decidable if, and only if, the well-known Star Problem in trace monoids is decidable too* [12,21].

1 Basic Notions

Let $(\mathbb{M}, \cdot)$ be a monoid with unit 1. For any subsets $\mathcal{L}$ and $\mathcal{L}'$ of $\mathbb{M}$, the *product* of $\mathcal{L}$ by $\mathcal{L}'$ is $\mathcal{L} \cdot \mathcal{L}' = \{x \cdot x' \mid x \in \mathcal{L} \wedge x' \in \mathcal{L}'\}$. We let $\mathcal{L}^0 = \{1\}$ and for any $n \in \mathbb{N}$, $\mathcal{L}^{n+1} = \mathcal{L}^n \cdot \mathcal{L}$; then $\mathcal{L}^\star = \bigcup_{n \in \mathbb{N}} \mathcal{L}^n$ is the *iteration* of $\mathcal{L}$. A language $\mathcal{L} \subseteq \mathbb{M}$ is *finitely generated* if there is a finite subset $\mathcal{L}_0$ of $\mathbb{M}$ such that $\mathcal{L} \subseteq \mathcal{L}_0^\star$. A subset of $\mathbb{M}$ is *rational* if it can be obtained from the finite subsets of $\mathbb{M}$ by means of unions, products and iterations. Any rational language is finitely generated. A subset $\mathcal{L}$ of $\mathbb{M}$ is *recognizable* if there exists a finite monoid $\mathbb{M}'$ and a monoid morphism $\eta : \mathbb{M} \to \mathbb{M}'$ such that $\mathcal{L} = \eta^{-1} \circ \eta(\mathcal{L})$. Equivalently, $\mathcal{L}$ is recognizable if and only if there exists a finite $\mathbb{M}$-automaton recognizing $\mathcal{L}$ — because the collection of all sets $\mathcal{L}/x = \{y \in \mathbb{M} \mid x \cdot y \in \mathcal{L}\}$ is finite. Thus, the set of recognizable subsets of any monoid is closed under union, intersection and complement.

A *pomset* over an alphabet Σ is a triple $t = (E, \preccurlyeq, \xi)$ where $(E, \preccurlyeq)$ is a finite partial order and ξ is a mapping from E to Σ. We denote by $\mathbb{P}(\Sigma)$ the class of all pomsets over Σ. Let $t = (E, \preccurlyeq, \xi)$ be a pomset and x, $y \in E$. Then y *covers* x (denoted $x \longrightarrow\!\!\!\prec y$) if $x \prec y$ and $x \prec z \preccurlyeq y$ implies $y = z$. The elements x and y are *concurrent* or *incomparable* if $\neg(x \preccurlyeq y) \wedge \neg(y \preccurlyeq x)$. A pomset $t = (E, \preccurlyeq, \xi)$ is *without auto-concurrency* if $\xi(x) = \xi(y)$ implies ($x \preccurlyeq y$ or $y \preccurlyeq x$) for all $x, y \in E$. A pomset can be seen as an abstraction of an execution of a concurrent system [32]. In this view, the elements e of E are *events* and their label $\xi(e)$ describes the basic action of the system that is performed by the event. Furthermore, the order describes the causal dependence between the events. In particular, if two events are concurrent, they can be executed in any order or even in parallel. A pomset is without auto-concurrency if no action can be performed concurrently with itself. An *ideal* of a pomset $t = (E, \preccurlyeq, \xi)$ is a subset $H \subseteq E$ such that $x \in H \wedge y \preccurlyeq x \Rightarrow y \in H$. For all $z \in E$, we denote by $\downarrow_t z$ the ideal of events below z, i.e. $\downarrow_t z = \{y \in E \mid y \preccurlyeq z\}$. If H is a subset of E, we denote by $\#_t^a(H)$ the number of events $x \in H$ such that $\xi(x) = a$. (We will omit the subscript t when it is clear from the context.) An *order extension* of a pomset $t = (E, \preccurlyeq, \xi)$ is a pomset $t' = (E, \preccurlyeq', \xi)$ such that $\preccurlyeq \subseteq \preccurlyeq'$. A *linear extension* of t is an order extension that is linearly ordered. Linear extensions of a pomset $t = (E, \preccurlyeq, \xi)$ can naturally be regarded as words over Σ. By $\mathrm{LE}(t) \subseteq \Sigma^\star$, we denote the set of linear extensions of a pomset t over Σ. Clearly, two isomorphic pomsets admit the same linear extensions. Noteworthy the converse property holds for pomsets without auto-concurrency: two pomsets without auto-concurrency t and t' are isomorphic iff $\mathrm{LE}(t) = \mathrm{LE}(t')$. For any subclass $\mathcal{L}$ of $\mathbb{P}(\Sigma)$, $\mathrm{LE}(\mathcal{L})$ denotes $\bigcup_{t \in \mathcal{L}} \mathrm{LE}(t)$.

Mazurkiewicz Traces. Let us now recall some basic notions of trace theory [9]. The concurrency of a distributed system is often represented by an *independence relation* over the alphabet of actions Σ, that is a binary, symmetric and irreflexive relation $\parallel \subseteq \Sigma \times \Sigma$. The associated *trace equivalence* is the least congruence $\sim$ over $\Sigma^\star$ such that $\forall a, b \in \Sigma, a \parallel b \Rightarrow ab \sim ba$. A trace $[u]$ is the equivalence class of a word $u \in \Sigma^\star$. We denote by $\mathbb{M}(\Sigma, \parallel)$ the set of all traces w.r.t. $(\Sigma, \parallel)$. Traces can easily be composed in the following way: $[u] \cdot [v] = [u.v]$. Then $\mathbb{M}(\Sigma, \|)$ appears as a monoid with the empty trace $[\varepsilon]$ as unit. A *trace language* is a subset $\mathcal{L} \subseteq \mathbb{M}(\Sigma, \|)$. It is easy to see that a trace language $\mathcal{L}$ is recognizable in $\mathbb{M}(\Sigma, \|)$ iff the set of associated linear extensions $\mathrm{LE}(\mathcal{L})$ is recognizable in the free monoid $\Sigma^\star$. Let $u \in \Sigma^\star$; then the trace $[u]$ is precisely the set of linear extensions $\mathrm{LE}(t)$ of a unique pomset $t = (E, \preccurlyeq, \xi)$ without auto-concurrency, that is, $[u] = \mathrm{LE}(t)$. Moreover t satisfies the following additional properties [24]:

MP_1: for all events $e_1, e_2 \in E$ with $\xi(e_1) \not\| \xi(e_2)$, we have $e_1 \preccurlyeq e_2$ or $e_2 \preccurlyeq e_1$;

MP_2: for all events $e_1, e_2 \in E$ with $e_1 \prec\!\!\cdot\, e_2$, we have $\xi(e_1) \not\| \xi(e_2)$.

Conversely any pomset satisfying these two axioms is a pomset without auto-concurrency whose linear extensions form a trace of $\mathbb{M}(\Sigma, \|)$. Thus one usually identifies $\mathbb{M}(\Sigma, \|)$ with the class of pomsets satisfying MP_1 and MP_2 — up to isomorphisms. The product of traces can now be viewed as a concatenation of pomsets: let $t_1 = (E_1, \preccurlyeq_1, \xi_1)$ and $t_2 = (E_2, \preccurlyeq_2, \xi_2)$ be two traces over $(\Sigma, \|)$; the concatenation $t_1 \cdot t_2$ is the pomset $t = (E_1 \uplus E_2, \preccurlyeq, \xi_1 \cup \xi_2)$ where $\preccurlyeq$ is the transitive closure of $\preccurlyeq_1 \;\cup\; \preccurlyeq_2 \;\cup\; \{(e_1, e_2) \in E_1 \times E_2 \mid \xi_1(e_1) \not\| \xi_2(e_2)\}$.

Basic Message Sequence Charts. MSCs are defined by several recommendations that indicate how one should represent them graphically [20]. More formally, they can be seen as particular labelled partial orders. Similar approaches can be traced to Lamport's diagrams [23] or Nielsen, Plotkin & Winskel's elementary event structures [29].

Let $\mathcal{I}$ be a finite set of processes, also called *instances*. For any instance $i \in \mathcal{I}$, Σ_i^{int} denotes a finite set of *internal actions*; the alphabet Σ_i is then the disjoint union of the set of *send actions* $\Sigma_i^! = \{i!j \mid j \in \mathcal{I} \setminus \{i\}\}$, the set of *receive actions* $\Sigma_i^? = \{i?j \mid j \in \mathcal{I} \setminus \{i\}\}$ and the set of internal actions Σ_i^{int}. We shall assume that the alphabets Σ_i are disjoint and we let $\Sigma_\mathcal{I} = \bigcup_{i \in \mathcal{I}} \Sigma_i$. Given an action $a \in \Sigma_\mathcal{I}$, we denote by $\mathrm{Ins}(a)$ the unique instance i such that $a \in \Sigma_i$, that is the particular instance on which each occurrence of action a occurs. Finally, for any pomset $(E, \preccurlyeq, \xi)$ over $\Sigma_\mathcal{I}$ we denote by $\mathrm{Ins}(e)$ the instance on which the event $e \in E$ occurs : $\mathrm{Ins}(e) = \mathrm{Ins}(\xi(e))$.

Definition 1.1. *A* basic message sequence chart (or basic MSC) *is a pomset* $M = (E, \preccurlyeq, \xi)$ *over* $\Sigma_\mathcal{I}$ *such that*

M_1: $\forall e, f \in E$: $\mathrm{Ins}(e) = \mathrm{Ins}(f) \Rightarrow (e \preccurlyeq f \vee f \preccurlyeq e)$

M_2: $\#^{i!j}(E) = \#^{j?i}(E)$ *for any distinct instances* i *and* j

M_3: $\big(\xi(e) = i!j \wedge \xi(f) = j?i \wedge \#^{i!j}(\downarrow e) = \#^{j?i}(\downarrow f)\big) \Rightarrow e \preccurlyeq f$

M_4: $[e \prec\!\!\cdot\, f \wedge \mathrm{Ins}(e) \neq \mathrm{Ins}(f)]$
$$\Rightarrow \big[\xi(e) = i!j \wedge \xi(f) = j?i \wedge \#^{i!j}(\downarrow e) = \#^{j?i}(\downarrow f)\big].$$

By M_1, events occurring on the same instance are linearly ordered : hence non-deterministic choice cannot be described within an MSC. Condition M_2 makes sure that there are as many send events from i to j than receive events from j to i; this expresses the reliability of the channels. Since the latter are assumed to be FIFO, the n-th message sent from i to j is received when the n-th event $j?i$ occurs; thus M_3 formalizes simply that the reception of any message will occur

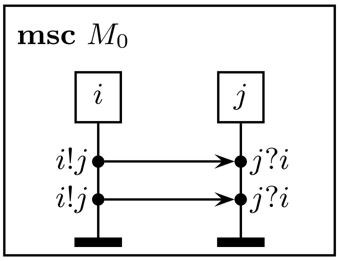

Fig. 1. A basic MSC

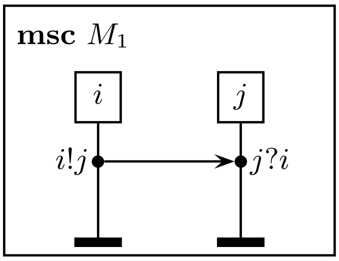

Fig. 2. $M_0 = M_1 \cdot M_1$

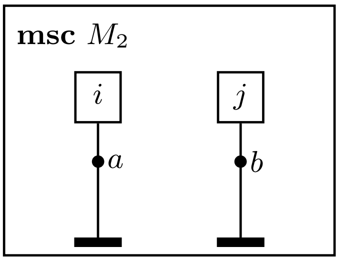

Fig. 3. Internal actions

only after the corresponding send event. Finally, by M_4, causality in M consists only in the linear dependency over each instance and the ordering of pairs of corresponding send and receive events.

Thus, basic MSCs are precisely the model studied in [1,14,7,16,17,25] although some authors forbid internal actions. Opposite to general sets of pomsets [22], basic MSCs satisfy a fundamental property known for Mazurkiewicz traces and also often considered in their generalization as P-traces [2,19].

Lemma 1.2. *Let M and M' be two basic MSCs. If* $\text{LE}(M) \cap \text{LE}(M') \neq \emptyset$ *then M and M' are isomorphic.*

Examples of basic MSCs over two instances i and j are described informally in Figures 1 and 2. There, each arrow $i!j \to j?i$ represents a pair of send-receive related events (and as usual MSCs should be read from top to bottom). Thus the MSC M_0 has two linear extensions $i!j.i!j.j?i.j?i$ and $i!j.j?i.i!j.j?i$. Hence the second send event and the first receive event are concurrent (or independent). The MSC M_1 has only one linear extension $i!j.j?i$ and thus does not describe actually any concurrent behavior.

Hierarchical Message Sequence Charts. We denote by b$\mathbb{MSC}$ the set of (isomorphism classes) of basic MSCs. The *asynchronous concatenation* of two basic MSCs $M_1 = (E_1, \preccurlyeq_1, \xi_1)$ and $M_2 = (E_2, \preccurlyeq_2, \xi_2)$ is $M_1 \cdot M_2 = (E, \preccurlyeq, \xi)$ where $E = E_1 \uplus E_2$, $\xi = \xi_1 \cup \xi_2$ and the partial order $\preccurlyeq$ is the transitive closure of $\preccurlyeq_1 \cup \preccurlyeq_2 \cup \{(e_1, e_2) \in E_1 \times E_2 \mid \text{Ins}(e_1) = \text{Ins}(e_2)\}$. It is easy to check that the asynchronous concatenation of two basic MSCs is a basic MSC. With Lemma 1.2, this concatenation can be shown to be associative and admits the empty MSC $(\emptyset, \emptyset, \emptyset)$ as unit. Therefore we shall refer to b$\mathbb{MSC}$ as the *monoid of basic message sequence charts.* As observed in [7], b$\mathbb{MSC}$ can also be viewed as a sub-monoid of the product $\prod_{i \in \mathcal{I}} \Sigma_i^\star$ provided with the component-wise concatenation. Numerous indecidability results were derived from this simple reduction [7, Th. 5].

The monoidal structure of basic MSCs enables us to use hierarchical specifications for sets of MSCs by composing finite languages by unions, concatenations or iterations — as this is usually done when considering rational languages within a monoid. In that way, we obtain hierarchical message sequence charts.

Definition 1.3. *A* hierarchical message sequence chart (HMSC) *is a rational expression of* $\mathrm{b}\mathbb{MSC}$*, that is, an expression built from basic MSCs by use of union, product and iteration.*

Note that the language of basic MSCs associated to an HMSC is finitely generated w.r.t. $\mathrm{b}\mathbb{MSC}$. We follow here the approach adopted, e.g., in [1,7,16,17,25] where HMSCs are however often flattened into message sequence graphs.

2 Channel-Bounded vs. Regular Languages

In this section, we show how one can decide whether a given HMSC does not induce divergence into channels, i.e. it could be implemented with channels having a finite capacity. This interesting property is called *channel-boundedness.*

Channel-Boundedness, Regularity, and Mazurkiewicz Traces. In [4], Ben-Abdallah & Leue defined the process divergence of HMSC by the existence of an infinite sequential execution that induce unbounded numbers of messages in channels. More recently, a simpler notion of boundedness was considered in [16,17,25] for the study of regular languages.

Definition 2.1. *The* channel-width *of a basic MSC M is*

$$\max_{i,j\in\mathcal{I},i\neq j}\{\#^{i!j}(H)-\#^{j?i}(H)\mid H \textit{ ideal of } M\}.$$

A language $\mathcal{L}\subseteq\mathrm{b}\mathbb{MSC}$ is channel-bounded *by an integer B if each basic MSC of $\mathcal{L}$ has a channel-width at most B.*

Recall now that graphical representations of MSCs should be read from top to bottom on each instance. Thus in the basic MSC M_3 of Fig. 4, the event labelled $k?j$ occurs before the event labelled $k!l$. Now the channel-width of M_3 of is 4. Removing the two events labelled $l!i$ and $i?l$ from M_3 would lead to a basic MSC with channel-width 5. Note also that the channel-width of $M_3\cdot M_3$ is 5. Consider again the basic MSC M_1 of Fig. 2; the rational language $\{M_1\}^\star$ is not channel-bounded.

As explained in the introduction, a particularly interesting notion of regularity was introduced in [16] and related to MSO logic [17] and message passing automata [25].

Definition 2.2. *A language $\mathcal{L}$ of basic MSCs is* regular *if its set of linear extensions* $\mathrm{LE}(\mathcal{L})=\bigcup_{M\in\mathcal{L}}\mathrm{LE}(M)$ *is recognizable in the free monoid $\Sigma_{\mathcal{I}}^\star$.*

We remark that regularity differs from recognizability. On one hand, any regular language is recognizable. But the converse fails: consider for instance again the MSC M_1 of Figure 2: the rational language $\{M_1\}^\star$ is recognizable in $\mathrm{b}\mathbb{MSC}$ but not regular. Actually, as observed in [16, Prop. 2.1], any regular language is channel-bounded.

From a remark of [17,25] and Lemma 1.2, it follows that sets of basic MSCs can be seen as generalized trace languages [30] or CCI sets of P-traces [2,19]. Therefore, regular sets of MSCs can be represented by recognizable subsets of Mazurkiewicz traces by means of a relabeling [2,19]. But since MSCs are very particular P-traces this basic relationship can be established here as follows.

Lemma 2.3 (D. Kuske). *Let B be a positive integer. We consider the finite alphabet $\Sigma = \Sigma_{\mathcal{I}} \times [0, B]$ and the independence relation $\| \subseteq \Sigma \times \Sigma$ such that*

$$(a, n) \not\| (a', n') \text{ if } \mathrm{Ins}(a) = \mathrm{Ins}(a') \text{ or } [\ \{a, a'\} = \{i!j, j?i\} \wedge n = n'\ ].$$

Let π_1 be the first projection from Σ to $\Sigma_{\mathcal{I}}$ which sends $(a, n) \in \Sigma$ to $a \in \Sigma_{\mathcal{I}}$. The map π_1 extends naturally to a map from the pomsets over Σ to the pomsets over $\Sigma_{\mathcal{I}}$ for which $(E, \preccurlyeq, \xi)$ is associated to $(E, \preccurlyeq, \pi_1 \circ \xi)$. Then for any basic MSC M with channel-width at most B, there exists a Mazurkiewicz trace $t \in \mathbb{M}(\Sigma, \|)$ such that $\pi_1(t) = M$.

Proof. Let $M = (e, \preccurlyeq, \xi)$ be an MSC with channel-width at most B. We easily check that the pomset $t = (E, \preccurlyeq, \xi_t)$ over Σ such that $\xi_t(e) = (\xi(e), \#_M^{\xi(e)}(\downarrow e) \bmod (B + 1))$ is a trace over $(\Sigma, \|)$. ■

By [17], any regular language of MSCs is MSO definable. Therefore we can use Büchi's Theorem for Mazurkiewicz traces [11,34] together with Lemma 2.3 to derive the following representation result.

Corollary 2.4. *Let $\mathcal{L}$ be a language of basic MSCs channel-bounded by B. With the notations of Lemma 2.3, let $\pi_1^{-1}(\mathcal{L})$ be the set of traces $t \in \mathbb{M}(\Sigma, \|)$ such that $\pi_1(t) \in \mathcal{L}$. If $\mathcal{L}$ is regular then $\pi_1^{-1}(\mathcal{L})$ is recognizable in $\mathbb{M}(\Sigma, \|)$ and $\pi_1 \circ \pi_1^{-1}(\mathcal{L}) = \mathcal{L}$.*

Thus any regular language of basic MSCs can be seen as a recognizable set of traces up to an adequate relabeling; moreover the latter depends on an upper bound for the channel-widths of the MSCs.

Easy Checking of the Channel-Boundedness Property. As established in [16, Th. 4.6], one cannot decide whether a rational language of b$\mathbb{MSC}$ is regular. In other words:

Theorem 2.5. *[16] It is undecidable to check whether the language of basic MSCs associated to a given HMSC is regular.*

In this section, we shall cope with this negative result in two different ways. A first natural approach is to weaken the problem: since any regular language is channel-bounded, one could aim at checking only the channel-boundedness of the language associated to an HMSC, instead of its regularity. This will be achieved by Theorem 2.8 and Corollary 2.9 below. Another way to deal with Theorem 2.5 is to look for a subclass of HMSCs that describes only regular languages. This is achieved in particular by sc-HMSCs defined below (Def. 2.11 and Cor. 2.13).

In order to represent a channel-bounded language of MSCs by a trace language, Lemma 2.3 indicates that it suffices to compute an upper bound for the

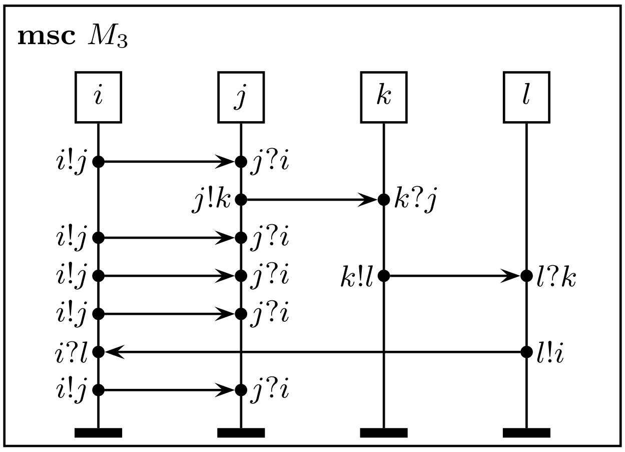

Fig. 4. A strongly connected MSC

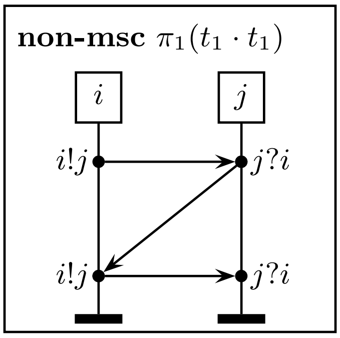

Fig. 5. $\pi_1(t_1 \cdot t_1) \neq M_0$

channel-width of the MSCs. First, if $\mathcal{L}_1$ and $\mathcal{L}_2$ are channel-bounded by B_1 and B_2 respectively then $\mathcal{L}_1 \cup \mathcal{L}_2$ is channel-bounded by $\max(B_1, B_2)$. For the product, we simply observe:

Lemma 2.6. *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two languages of basic MSCs channel-bounded by B_1 and B_2 respectively. Then $\mathcal{L}_1 \cdot \mathcal{L}_2$ is channel-bounded by $B_1 + B_2$.*

Thus, the main problem consists in deciding whether the iteration of a channel-bounded language is channel-bounded *and to compute a bound.* For later purposes, it is convenient to slightly extend now a useful notion related to MSCs.

Definition 2.7. *The* communication graph *of a basic MSC $M = (E, \preccurlyeq, \xi)$ is the directed graph $(\mathcal{I}_M, \mapsto)$ where $\mathcal{I}_M$ is the set of* active instances *of M: $\mathcal{I}_M = \{i \in \mathcal{I} \mid \exists e \in E, \mathrm{Ins}(e) = i\}$, and such that $(i,j) \in \mapsto$ if there is $e \in E$ such that $\xi(e) = i!j$.*

Thus there is an edge from i to j if M specifies a communication from i to j. In [1,28,16], an *extended communication graph* $(\mathcal{I}, \mapsto)$ is considered without restriction to the active instances, but still with the same edges. Actually our slight variation is meant to cope with *internal actions* in Cor. 2.13.

The communication graph is the basis of a useful criterion to check whether the iteration of a language of basic MSCs is channel-bounded.

Theorem 2.8. *Let $\mathcal{L} \subseteq \mathrm{b}\mathbb{MSC}$. The following conditions are equivalent:*
(i) *$\mathcal{L}^\star$ is channel-bounded.*
(ii) *$\mathcal{L}$ is channel-bounded and the communication graph of each $M \in \mathcal{L}$ is* locally strongly connected *— i.e. each connected component is strongly connected.*
Moreover, if (ii) *holds and if $\mathcal{L}$ is channel-bounded by B then $\mathcal{L}^\star$ is channel-bounded by $2^{N^2}.(N+1).B$ where $N = \mathrm{Card}(\mathcal{I})$.*

Proof. First we prove (i) ⇒ (ii) by contradiction. We assume that $\mathcal{L}$ contains a basic MSC M for which at least one connected component of its communication graph is not strongly connected. This means that there are two distinct instances

i and j such that $i \mapsto j$ and there is no path from j to i. We simply observe here that the channel-width of M^{B+2} is larger than $B+1$.

We prove now (ii) ⇒ (i). Let i and j be two fixed distinct instances. We shall use several time the following observation:

Claim. *Let M_1,..., M_n be n basic MSCs of $\mathcal{L}$ and $M = M_1 \cdot ... \cdot M_n$. Let K be the set of integers $k \in [1,n]$ such that there is an edge $i \mapsto j$ in the communication graph of M_k. Let $k_1 < k_2$ be two integers of $[1,n]$ and let $e \in E_{k_2}$ be such that $\xi(e) = i!j$. If $\forall f \in E, [f \in E_k \wedge f \preccurlyeq e \wedge \xi(f) = j?i] \Rightarrow k < k_1$ then* Card$(K \cap [k_1,k_2]) \leqslant 2^{N.(N-1)}.(N+1)$. □

Let us prove this claim first, by contradiction. We assume that Card$(K \cap [k_1,k_2]) > 2^{N.(N-1)}.(N+1)$. Since there are only $2^{N.(N-1)}$ distinct extended communication graphs, there are in the family $(M_k)_{k \in K \cap [k_1,k_2]}$ at least $N+1$ MSCs with the same communication graph G_0. The graph G_0 contains an edge (i,j). Since any connected component of G_0 is strongly connected, there is a path $j = i_1 \mapsto i_2 \mapsto ... \mapsto i_r = i$ in G_0 with $r \leqslant \text{Card}(\mathcal{I}) = N$. We denote by J the set of integers $k \in K \cap [k_1,k_2]$ such that the communication graph of M_k is G_0. Then Card$(J) \geqslant N+1$. Let $j_1, j_2, ..., j_{\text{Card}(J)}$ be an increasing enumeration of J. Let f be an event of M_{j_1} such that $\xi(f) = j?i$. Since Card$(J) - 1 \geqslant \text{Card}(\mathcal{I}) \geqslant r$, there is an event g in $M_{\text{Card}(J)-1}$ such that Ins$(g) = i$ and $f \preccurlyeq g$. Now $g \preccurlyeq e$ hence $f \preccurlyeq e$. This contradicts $f \in E_{k_1}$.

We consider now some MSCs M_1,..., M_n in $\mathcal{L}$ and their concatenation $M = (E, \preccurlyeq, \xi) = M_1 \cdot ... \cdot M_n$. We let K be as above. Let $e_0 \in E$ be such that $\xi(e_0) = i!j$ and consider k_0 to be the integer of $[1,n] \cap K$ such that $e_0 \in E_{k_0}$. It is sufficient to show that $\#^{i!j}_M(\downarrow_M e_0) - \#^{j?i}_M(\downarrow_M e_0) \leqslant 2^{N^2}.(N+1).B$.

1. We assume first that Card$(K \cap [1,k_0]) \leqslant 2^{N.(N-1)}.(N+1)$. We let $M' = M_1 \cdot ... \cdot M_{k_0}$. We observe that $\downarrow_M e_0 = \downarrow_{M'} e_0$. By Lemma 2.6, $\#^{i!j}_M(\downarrow_M e_0) - \#^{j?i}_M(\downarrow_M e_0) = \#^{i!j}_{M'}(\downarrow_{M'} e_0) - \#^{j?i}_{M'}(\downarrow_{M'} e_0) \leqslant 2^{N.(N-1)}.(N+1).B$.
2. We assume now that Card$(K \cap [1,k_0]) > 2^{N.(N-1)}.(N+1)$. Then, according to the claim above, there are some events $f \in E$ such that $\xi(f) = j?i$ and $f \preccurlyeq e_0$. Among all these events below e_0 and labelled $j?i$, we consider f_1 to be the maximal one. Let k_1 be the integer such that $f_1 \in E_{k_1}$. Since M_{k_1} is basic, there is an event $e_1 \in E_{k_1}$ such that $\xi(e_1) = i!j$ and $\#^{i!j}_{M_{k_1}}(\downarrow e_1) = \#^{j?i}_{M_{k_1}}(\downarrow f_1)$. Therefore $\#^{i!j}_M(\downarrow e_1) = \#^{j?i}_M(\downarrow f_1)$. According to the claim above, Card$(K \cap [k_1+1, k_0]) \leqslant 2^{N.(N-1)}.(N+1)$ — otherwise f_1 is not maximal. We consider $k' \in K \cap [k_1+1, k_0-1]$. Let $e \in E_k$ be such that $\xi(e) = i!j$. Then there is no event $f \in E_k$ such that $f \preccurlyeq_{M_k} e$ and $\xi(f) = j?i$ (otherwise $f \preccurlyeq e_0$ and f_1 is not maximal). Therefore $\#^{i!j}_{M_k}(E_k) \leqslant B$ because the channel-width of M_k is at most B. Similarly, Card$\{e \in E_{k_1} \mid \xi(e) = i!j \wedge e_1 \prec e\} \leqslant B$ and $\#^{i!j}_{M_{k_0}}(\downarrow_{M_{k_0}} e_0) \leqslant B$. Consequently, $\#^{i!j}_M(\downarrow_M e_0) \leqslant (2 + 2^{N.(N-1)}.(N+1)).B + \#^{i!j}_M(\downarrow_M e_1)$. Hence $\#^{i!j}_M(\downarrow_M e_0) - \#^{j?i}_M(\downarrow_M e_0) \leqslant (2 + 2^{N.(N-1)}.(N+1)).B$.

■

Now, the product or union of two languages $\mathcal{L}_1$ and $\mathcal{L}_2$ is channel-bounded if, and only if, $\mathcal{L}_1$ and $\mathcal{L}_2$ are channel-bounded. For the iteration, we observe that one can inductively compute the set of all communication graphs of all the MSCs associated to a given HMSC. Thus channel-boundedness is easily decidable:

Corollary 2.9. *An HMSC is channel-bounded iff iteration occurs only over sets of MSCs whose communication graphs are locally strongly connected.*

As a consequence, a *rational* language of MSCs is channel-bounded iff it is divergence-free in the sense of [4].

First Application to Regular Languages. We are here interested in a subclass of HMSCs that describes only regular languages. Again, union and product do not raise problems at all:

Lemma 2.10. *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two regular languages of basic MSCs. Then $\mathcal{L}_1 \cup \mathcal{L}_2$ and $\mathcal{L}_1 \cdot \mathcal{L}_2$ are regular too.*

Now, a rather simple way to ensure the regularity of languages associated to hierarchical message sequence charts is to restrict to sc-HMSCs. This restriction is actually a reformulation of a condition of “local synchronization” or “boundedness” introduced in the framework of message sequence graphs [28,1].

Definition 2.11. *A hierarchical MSC is an* sc-HMSC *if iteration occurs only over sets of MSCs whose communication graphs are strongly connected.*

We show here that this restriction corresponds precisely to an approach previously followed by Ochmański in the framework of Mazurkiewicz traces [31]. Recall that a trace $t \in \mathbb{M}(\Sigma, \|)$ is *connected* if the restriction of the dependence graph $(\Sigma, \not\|)$ to the subset of actions appearing in t is connected. Then a subset of $\mathbb{M}(\Sigma, \|)$ is *c-rational* if it can be obtained from finite subsets by means of unions, products and iterations over subsets of connected traces.

Theorem 2.12. *[31] A trace language is recognizable iff it is c-rational.*

The next result was originally shown in [28,1] and [16] however under the assumption that there is no internal action. The (restricted) communication graph of Def. 2.7 enables us to extend this relationship in the more general present setting. We also show how it can be inferred from Theorem 2.12.

Corollary 2.13. *Let $\mathcal{L}$ be a* finitely generated *language of basic MSCs. Then $\mathcal{L}$ is regular if, and only if, it is the language of an sc-HMSC.*

Proof. We consider first a regular, finitely generated language $\mathcal{L}$ of basic MSCs. We consider Σ to be a finite family of basic MSCs such that $\mathcal{L} \subseteq \Sigma^\star \subseteq \text{b}\mathbb{MSC}$. We may assume that the communication graph of each MSC $M \in \Sigma$ is connected. Since $\mathcal{L}$ is recognizable, there is a deterministic finite full b$\mathbb{MSC}$-automaton $\mathcal{A} = (Q, \{\imath\}, \longrightarrow, F)$ that recognizes $\mathcal{L}$. Let $\mathcal{A}_0 = (Q, \{\imath\}, \longrightarrow_0, F)$ be the automaton over the alphabet Σ such that $\xrightarrow{M}_0 = \xrightarrow{M} \cap (Q \times \Sigma \times Q)$. We consider the independence relation over Σ such that $M \| M'$ if for all events e of M and for all events e' of M', $\text{Ins}(e) \neq \text{Ins}(e')$. Then, if $q \xrightarrow{M}_0 q_1 \xrightarrow{M'}_0 q_2$ and $M \| M'$

then there is a state q_3 such that $q \xrightarrow{M'}_0 q_3 \xrightarrow{M} q_2$ because $M \cdot M' = M' \cdot M$ and $\mathcal{A}$ is deterministic. The language $\mathcal{L}_0$ recognized by $\mathcal{A}_0$ is recognizable in the free monoid $\Sigma^\star$; it is also closed for the commutation of independent MSCs. Therefore $\mathcal{L}_0$ can be identified to a recognizable trace language of $\mathbb{M}(\Sigma, \|)$. By Th. 2.12, we can consider a c-rational expression h that describes $\mathcal{L}_0 \in \mathbb{M}(\Sigma, \|)$. We can see also h as a rational expression over bMSC — that is, as an HMSC — that describes actually $\mathcal{L}$. Recall now that the communication graph of each MSC $M \in \Sigma$ is connected; moreover the star operation is taken in h only over sets of connected traces of $\mathbb{M}(\Sigma, \|)$. Therefore the star operation is taken in h over sets of MSCs which are connected. Now we know that $\mathcal{L}$ is regular, hence channel-bounded. Therefore Th. 2.8 ensures that the star operation in h is only taken over sets of MSCs which are *strongly* connected, i.e. h is an sc-HMSC.

For the converse, Lemma 2.10 shows it is sufficient to consider the iteration of a regular language $\mathcal{L}_0$ of basic MSCs. Then $\mathcal{L}_0$ is channel-bounded by some integer B_0. By Th. 2.8, the language $\mathcal{L}_0^\star$ is channel-bounded by $B = 2^{N^2}.(N+1).B_0$ where $N = \text{Card}(\mathcal{I})$. We use here Lemma 2.3 with $\Sigma = \Sigma_{\mathcal{I}} \times [0, B]$. Then $\pi_1^{-1}(\mathcal{L}_0)$ is recognizable in $\mathbb{M}(\Sigma, \|)$ (Corollary 2.4). Moreover $\pi_1^{-1}(\mathcal{L}_0)$ is connected because each MSC of $\mathcal{L}_0$ consists of strongly connected MSCs. Therefore $\mathcal{L}^\dagger = \pi_1^{-1}(\mathcal{L}_0)^\star$ is recognizable too (Th. 2.12). Consider now the language $\mathcal{L}_B \subseteq \mathbb{M}(\Sigma, \|)$ that consists of the traces t such that $\pi_1(t)$ is a basic MSC of channel-width at most B. We can show that $\mathcal{L}_B$ is definable in MSO logic, hence it is recognizable in $\mathbb{M}(\Sigma, \|)$ [34]. Then $\mathcal{L}^\dagger \cap \mathcal{L}_B$ is recognizable in $\mathbb{M}(\Sigma, \|)$. Consequently, the set of linear extensions $L^\dagger = \text{LE}(\mathcal{L}^\dagger \cap \mathcal{L}_B)$ is recognizable in $\Sigma^\star$ and its image through $\pi_1 : \Sigma^\star \to \Sigma_{\mathcal{I}}^\star$ is a recognizable language of $\Sigma_{\mathcal{I}}^\star$. To conclude we can show that $\mathcal{L}^\dagger \cap \mathcal{L}_B = \pi_1^{-1}(\mathcal{L}_0^\star)$ hence $\pi_1(L^\dagger) = \text{LE}(\mathcal{L}_0^\star)$. ■

Note finally that the restriction of communication graphs to active instances makes sense: with M_2 of Fig. 3, $\{M_2\}^\star$ is obviously not regular.

3 Connecting Two Star Problems

The subclass of sc-HMSCs describes precisely all regular languages of MSCs (Corollary 2.13). However allowing iteration over sets of strongly connected MSCs only can be considered to be too restrictive. We investigate now how we could weaken this restriction while keeping the same expressive power. By Lemma 2.10, it suffices to forbid the iteration of a regular language whenever the resulting language is not regular. However, this might lead to an intractable criterion. Indeed, we shall prove here the following result.

Theorem 3.1. *Consider the two following problems:*

Pb_1*: Given a finite independence alphabet* $(\Sigma, \|)$ *and a recognizable language* $\mathcal{L}$ *of* $\mathbb{M}(\Sigma, \|)$*, decide whether* $\mathcal{L}^\star$ *is recognizable in* $\mathbb{M}(\Sigma, \|)$*.*

Pb_2*: Given a finite set of instances* $\mathcal{I}$ *and a regular language* $\mathcal{L}$ *of* bMSC *over* $\mathcal{I}$*, decide whether* $\mathcal{L}^\star$ *is regular.*

Then Pb_1 *is decidable if, and only if,* Pb_2 *is decidable.*

Recall that Pb_1 is known as "the Star Problem in trace monoids," and it is still an open question to know whether it is decidable [21]. The proof of Theorem 3.1 proceeds from Propositions 3.2 and 3.6 below.

It is well-known that some classical telecommunication protocols — such as the alternating bit protocol — cannot be described by HMSCs because they are not *finitely generated* languages. For this reason, compositional MSCs are introduced in [13]: they enable to describe any regular language of MSCs by a rational expression. That is why, we stress that *an important aspect of Theorem 3.1 is that we do not restrict to finitely generated languages in the statement of* Pb_2. In fact, Propositions 3.2 and 3.6 also show that Theorem 3.1 still holds if we restrict to finitely generated languages. First, the "if" part of Theorem 3.1.

PROPOSITION 3.2. *Consider the following variation of* Pb_2.

Pb'_2*: Given a finite set of instances* $\mathcal{I}$ *and a regular* finitely generated *language* $\mathcal{L}$ *of* $\mathrm{b}\mathbb{MSC}$ *over* $\mathcal{I}$*, decide whether* $\mathcal{L}^\star$ *is regular.*

If Pb'_2 *is decidable then the Star Problem* Pb_1 *of Theorem 3.1 is decidable too.*

Proof. Let $(\Sigma, \|)$ be a finite independence alphabet. There exists a finite set of instances $\mathcal{I}$ and a family of basic MSCs $(M_a)_{a\in\Sigma}$ with the following properties:

- $a\|b \Rightarrow M_a \cdot M_b = M_b \cdot M_a$;
- the morphism $\psi : \mathbb{M}(\Sigma, \|) \to \mathrm{b}\mathbb{MSC}$ such that $\psi(a) = M_a$ is *one-to-one.*
- for any recognizable language $\mathcal{L} \in \mathbb{M}(\Sigma, \|)$, given a finite automaton over Σ that recognizes $\mathrm{LE}(\mathcal{L})$, one can effectively build a finite automaton over $\Sigma_{\mathcal{I}}$ that recognizes $\mathrm{LE}(\psi(\mathcal{L}))$.

Note here that ψ is well-defined because $a\|b \Rightarrow M_a \cdot M_b = M_b \cdot M_a$. Moreover $\psi(\mathbb{M}(\Sigma, \|))$ is finitely generated. (See [15] for an example of such a family that was used to prove Th. 2.5).

We show that we can decide whether $\mathcal{L}_0^\star$ is recognizable in $\mathbb{M}(\Sigma, \|)$ when $\mathcal{L}_0$ is a recognizable language of $\mathbb{M}(\Sigma, \|)$ given by a finite automaton over Σ that recognizes $\mathrm{LE}(\mathcal{L}_0)$. We can effectively construct an automaton $\mathcal{A}'$ that recognizes $\mathrm{LE}(\psi(\mathcal{L}_0))$. Then $\psi(\mathcal{L}_0)$ is finitely generated. Thus, we need only to show that $\mathcal{L}_0^\star$ is recognizable if and only if $\psi(\mathcal{L}_0)^\star$ is regular. Assume first that $\mathcal{L}_0^\star$ is recognizable. Let $\mathcal{A}_0$ be a finite automaton over Σ that recognizes $\mathrm{LE}(\mathcal{L}_0^\star)$. Then there is a finite automaton $\mathcal{A}'$ over $\Sigma_{\mathcal{I}}$ that recognizes $\mathrm{LE}(\psi(\mathcal{L}_0^\star))$. Since ψ is a monoid morphism, $\psi(\mathcal{L}_0^\star) = \psi(\mathcal{L}_0)^\star$ hence $\psi(\mathcal{L}_0)^\star$ is regular. Conversely, assume that $\psi(\mathcal{L}_0)^\star$ is regular. Then $\mathcal{L}_0^\star = \psi^{-1}(\psi(\mathcal{L}_0)^\star)$ because ψ is one-to-one and $\psi(\mathcal{L}_0^\star) = \psi(\mathcal{L}_0)^\star$. Since $\psi(\mathcal{L}_0)^\star$ is regular, it is recognizable in $\mathrm{b}\mathbb{MSC}$ hence $\mathcal{L}_0^\star$ is recognizable in $\mathbb{M}(\Sigma, \|)$. ■

The other direction of Theorem 3.1 turns out to be more difficult and is, in our opinion, the most interesting part of this paper. The reason is that we cannot simply use Kuske's relabeling technique (Lemma 2.3) because the mapping π_1 does not preserves products: consider for instance MSC M_1 of Fig. 2 and let $t_1 \in \mathbb{M}(\Sigma, \|)$ be such that $\pi_1(t_1) = M_1$. Then t_1 is simply a word $(i!j, n).(j?i, n)$ for some $n \in \mathbb{N}$. We observe here that $\pi_1(t_1 \cdot t_1) \neq M_1 \cdot M_1$ (cf. Fig. 1 and 5).

To cope with this algebraic flaw, we shall adapt the representation technique as follows.

Definition 3.3. *Let B be a positive integer and let $(\Sigma, \|)$ be the corresponding independence alphabet defined in Lemma 2.3. We denote by $\rho : \mathbb{P}(\Sigma) \to \mathbb{P}(\Sigma_{\mathcal{I}})$ the function from the pomsets over Σ to the pomsets over $\Sigma_{\mathcal{I}}$ such that $t = (E, \preccurlyeq, \xi)$ maps to $(E, \preccurlyeq^{\dagger}, \pi_1 \circ \xi)$ where $\preccurlyeq^{\dagger}$ is the transitive closure of*

$$\{(e,f) \in E^2 \mid \xi(e) \not\parallel \xi(f) \wedge e \preccurlyeq f \wedge \forall i,j \in \mathcal{I}, (\pi_1(\xi(e)) \neq j?i \vee \pi_1(\xi(f)) \neq i!j)\}$$

For any language $\mathcal{L} \in$ b$\mathbb{MSC}$, we denote by $\rho^{-1}(\mathcal{L})$ the set of all traces $t \in \mathbb{M}(\Sigma, \|)$ such that $\rho(t) \in \mathcal{L}$.

We first observe that Lemma 2.3 yields

Corollary 3.4. *For any basic MSC M with channel-width at most B, there exists a Mazurkiewicz trace $t \in \mathbb{M}(\Sigma, \|)$ such that $\rho(t) = M$.*

Now the map $\rho : \mathbb{P}(\Sigma) \to \mathbb{P}(\Sigma_{\mathcal{I}})$ satisfies two crucial properties that are not fulfilled by $\pi_1 : \mathbb{P}(\Sigma) \to \mathbb{P}(\Sigma_{\mathcal{I}})$. First, $\rho^{-1}($b$\mathbb{MSC})$ is a sub-monoid of $\mathbb{M}(\Sigma, \|)$ and $\rho : \rho^{-1}($b$\mathbb{MSC}) \to$ b$\mathbb{MSC}$ is a monoid morphism. Second $\rho^{-1}(\mathcal{L}^{\star}) = \rho^{-1}(\mathcal{L})^{\star}$ for any language $\mathcal{L} \subseteq$ b$\mathbb{MSC}$. For this, it suffices to check the following property.

Lemma 3.5. *With the notations of Def. 3.3, let M_1 and M_2 be two basic MSCs and let t be a trace over $\mathbb{M}(\Sigma, \|)$ such that $\rho(t) = M_1 \cdot M_2$. Then there are two traces t_1 and t_2 such that $\rho(t_1) = M_1$, $\rho(t_2) = M_2$ and $t = t_1 \cdot t_2$.*

Proof. We consider $M_1 = (E_1, \preccurlyeq_1, \xi_1)$ and $M_2 = (E_2, \preccurlyeq_2, \xi_2)$ two basic MSCs. We may assume here that $E_1 \cap E_2 = \emptyset$. Let $t = (E, \preccurlyeq, \xi_t)$ be a trace over $\mathbb{M}(\Sigma, \|)$ such that $\rho(t) = M_1 \cdot M_2$. Then $E = E_1 \cup E_2$ and E_1 is an ideal of $\rho(t)$. Actually, the proof follows from the key observation that E_1 is an ideal of t as well. We proceed by contradiction. We can show that there are $e \in E_2$ and $f \in E_1$ such that $e \mathrel{-\!\!\!\prec}_t f$. Since t is a trace, $\xi(e) \not\parallel \xi(f)$. But $\neg(e \preccurlyeq_{\rho(t)} f)$ since E_1 is an ideal of $\rho(t)$. Therefore, $\xi_2(e) = j?i$ and $\xi_1(f) = i!j$. Now M_1 is a basic MSC so there exists an event $e_0 \in E_1$ such that $\xi(e_0) = j?i$ and $\#^{j?i}_{M_1}(\downarrow e_0) = \#^{i!j}_{M_1}(\downarrow f)$. This implies $\#^{j?i}_{M_1 \cdot M_2}(\downarrow e_0) = \#^{i!j}_{M_1 \cdot M_2}(\downarrow f)$. Hence $f \preccurlyeq_{\rho(t)} e_0$ because $M_1 \cdot M_2$ is also a basic MSC. But $e_0 \preccurlyeq_{\rho(t)} e$ because $e \in E_2$, $e_0 \in E_1$ and $\mathrm{Ins}(e_0) = \mathrm{Ins}(e)$. Therefore $f \preccurlyeq_{\rho(t)} e$. This contradicts $e \mathrel{-\!\!\!\prec}_t f$. ∎

Finally, similarly to π_1, we observe that $\rho^{-1}($b$\mathbb{MSC})$ is definable in MSO logic. From this we can adapt Corollary 2.4 to prove a third technical remark: for any regular language $\mathcal{L} \subseteq$ b$\mathbb{MSC}$, $\rho^{-1}(\mathcal{L})$ is recognizable in $\mathbb{M}(\Sigma, \|)$.

Proposition 3.6. *Consider the following variation of* Pb_1.

Pb_1'*: Given a finite independence alphabet* $(\Sigma, \|)$ such that each action appears in at most two maximal cliques of the dependence graph $(\Sigma, \not\parallel)$ *and a recognizable language $\mathcal{L}$ of $\mathbb{M}(\Sigma, \|)$, decide whether $\mathcal{L}^{\star}$ is recognizable.*

If Pb_1' *is decidable then* Pb_2 *of Theorem 3.1 is decidable too.*

Proof. Let $\mathcal{L}$ be a regular language of b$\mathbb{MSC}$ described by a finite automaton over $\Sigma_{\mathcal{I}}$ that recognizes $\mathrm{LE}(\mathcal{L})$. Let B be the number of states of $\mathcal{A}$. Then $\mathcal{L}$ is channel-bounded by B. Clearly, we can decide from $\mathcal{A}$ whether each connected component in the communication graph of all MSCs of $\mathcal{L}$ is strongly connected.

If this is not the case, then $\mathcal{L}^\star$ is not channel-bounded (Th. 2.8) hence not regular. Therefore, we can assume now that this is the case. Then, again by Th. 2.8, $\mathcal{L}^\star$ is channel-bounded by $B' = 2^{N^2}.(N+1).B$. We use now the notations of Def. 3.3 with $\Sigma = \Sigma_{\mathcal{I}} \times [0, B']$. Since $\mathcal{L}$ is regular, $\mathcal{L}_0 = \rho^{-1}(\mathcal{L})$ is recognizable in $\mathbb{M}(\Sigma, \|)$; moreover *an automaton recognizing* $\mathrm{LE}(\mathcal{L}_0)$ *can effectively be computed from* $\mathcal{A}$. To conclude this proof, we show that $\mathcal{L}_0^\star$ is recognizable in $\mathbb{M}(\Sigma, \|)$ if, and only if, $\mathcal{L}^\star$ is regular in bMSC. By the second observation, we have $\rho^{-1}(\mathcal{L}^\star) = \rho^{-1}(\mathcal{L})^\star = \mathcal{L}_0^\star$. Assume first that $\mathcal{L}^\star$ is regular; then (third observation) $\rho^{-1}(\mathcal{L}^\star)$ is recognizable in $\mathbb{M}(\Sigma, \|)$. Conversely, assume that $\mathcal{L}_0^\star$ is recognizable in $\mathbb{M}(\Sigma, \|)$. Then $\mathrm{LE}(\mathcal{L}_0^\star)$ is recognizable in $\Sigma^\star$ hence definable in MSO logic. We consider now lexicographic normal forms of pomsets over $\Sigma_{\mathcal{I}}$ with the condition that $j?i < i!j$ for any two distinct instances i and j. Since ρ is a morphism, $\rho(\mathcal{L}_0^\star) \subseteq \mathcal{L}^\star$, hence $\rho(\mathcal{L}_0^\star) = \mathcal{L}^\star$. Therefore $\mathcal{L}^\star$ is the set of basic MSCs M whose lexicographic normal forms belong to $\pi_1(\mathrm{LE}(\mathcal{L}_0^\star))$. Thus $\mathcal{L}^\star$ is MSO definable [10,11]. Since it is also channel-bounded, it is regular [17]. ■

Discussion. Another corollary of Prop. 3.2 and 3.6 is the following reduction of the Star Problem:

Corollary 3.7. *The Star Problem is decidable for all independence alphabets (*Pb_1 *of Theorem 3.1) if, and only if, it is decidable for all independence alphabets such that each action appears in at most 2 maximal cliques (*Pb_1' *of Prop. 3.6).*

To our knowledge, this reduction does not follow from known results. It remains however unclear to us whether this reduction could be useful to provide a new approach for an answer to this difficult question.

Acknowledgments. Many thanks to Dietrich Kuske for numerous motivating discussions on this subject and several suggestions to simplify the proofs and improve the presentation of the results.

References

1. Alur R. and Yannakakis M.: *Model Checking of Message Sequence Charts.* CONCUR'99, LNCS **1664** (1999) 114–129
2. Arnold A.: *An extension of the notion of traces and asynchronous automata.* Theoretical Informatics and Applications **25** (1991) 355–393
3. Ben-Abdallah H. and Leue S.: *Syntactic Analysis of Message Sequence Chart Specifications.* Technical report 96-12 (University of Waterloo, Canada, 1996)
4. Ben-Abdallah H. and Leue S.: *Syntactic Detection of Process Divergence and Non-local Choice in Message Sequence Charts.* TACAS'97, LNCS **1217** (1997) 259–274
5. Booch G., Jacobson I. and Rumbough J.: *Unified Modelling Language User Guide.* (Addison-Wesley, 1997)
6. Büchi J.R.: *Weak second-order arithmetic and finite automata.* Z. Math. Logik Grundlagen Math. **6** (1960) 66–92
7. Caillaud B., Darondeau Ph., Hélouët L. and Lesventes G.: *HMSCs as partial specifications... with PNs as completions.* Proc. of MOVEP'2k, Nantes (2000) 87–103
8. Cori R., Métivier Y. and Zielonka W.: *Asynchronous mappings and asynchronous cellular automata.* Information and Computation **106** (1993) 159–202

9. Diekert V. and Rozenberg G.: *The Book of Traces.* (World Scientific, 1995)
10. Droste M. and Kuske D.: *Logical definability of recognizable and aperiodic languages in concurrency monoids.* LNCS **1092** (1996) 233–251
11. Ebinger W. and Muscholl A.: *Logical definability on infinite traces.* Theoretical Comp. Science **154** (1996) 67–84
12. Gastin P., Ochmański E., Petit A. and Rozoy, B.: *On the decidability of the star problem.* Information Processing Letters **44** (1992) 65–71
13. Gunter E.L., Muscholl A. and Peled D.: *Compositional Message Sequence Charts.* TACAS 2001, LNCS (2001) – To appear.
14. Hélouët L., Jard C. and Caillaud B.: *An effective equivalence for sets of scenarios represented by HMSCs.* Technical report, PI-1205 (IRISA, Rennes, 1998)
15. Henriksen J.G, Mukund M., Narayan Kumar, K. and Thiagarajan P.S.: *Towards a theory of regular MSC languages.* Technical report (BRICS RS-99-52, 1999)
16. Henriksen J.G., Mukund M., Narayan Kumar K. and Thiagarajan P.S.: *On message sequence graphs and finitely generated regular MSC language.* LNCS **1853** (2000) 675–686
17. Henriksen J.G., Mukund M., Narayan Kumar K. and Thiagarajan P.S.: *Regular collections of message sequence charts.* MFCS 2000, LNCS **1893** (2000) 405–414
18. Holzmann G.J.: *Early Fault Detection.* TACAS'96, LNCS **1055** (1996) 1–13
19. Husson J.-Fr. and Morin R.: *On Recognizable Stable Trace Languages.* FoSSaCS 2000, LNCS **1784** (2000) 177–191
20. ITU-TS: *Recommendation Z.120: Message Sequence Charts.* (Geneva, 1996)
21. Kirsten D. and Richomme G.: *Decidability Equivalence Between the Star Problem and the Finite Power Problem in Trace Monoids.* Technical Report ISSN 1430-211X, TUD/FI99/03 (Dresden University of Technology, 1999)
22. Kuske D. and Morin R.: *Pomsets for Local Trace Languages: Recognizability, Logic and Petri Nets.* CONCUR 2000, LNCS **1877** (2000) 426–441
23. Lamport L.: *Time, Clocks and the Ordering of Events in a Distributed System.* Comm. of the ACM, vol. 21, N **27** (1978) – ACM
24. Mazurkiewicz A.: *Concurrent program schemes and their interpretations.* Aarhus University Publication (DAIMI PB-78, 1977)
25. Mukund M., Narayan Kumar K. and Sohoni M.: *Synthesizing distributed finite-state systems from MSCs.* CONCUR 2000, LNCS **1877** (2000) 521–535
26. Muscholl A., Peled D. and Su Z.: *Deciding Properties for Message Sequence Charts.* FoSSaCS'98, LNCS **1378** (1998) 226–242
27. Muscholl A.: *Matching Specifications for Message Sequence Charts.* FoSSaCS'99, LNCS **1578** (1999) 273–287
28. Muscholl A. and Peled D.: *Message sequence graphs and decision problems on Mazurkiewicz traces.* Proc. of MFCS'99, LNCS **1672** (1999) 81–91
29. Nielsen M., Plotkin G. and Winskel G.: *Petri nets, events structures and domains, part 1.* Relationships between Models of Concurrency, TCS **13** (1981) 85–108
30. Nielsen M., Sassone V. and Winskel G.: *Relationships between Models of Concurrency.* Rex'93: A decade of concurrency, LNCS **803** (1994) 425–475
31. Ochmański E.: *Regular behaviour of concurrent systems.* Bulletin of the EATCS **27** (Oct. 1985) 56–67
32. Pratt V.: *Modelling concurrency with partial orders.* Int. J. of Parallel Programming **15** (1986) 33–71
33. Sakarovitch J.: *The "last" decision problem for rational trace languages.* Proc. LATIN'92, LNCS **583** (1992) 460–473
34. Thomas W.: *On logical definability of trace languages.* Technical University of Munich, report TUM-I9002 (1990) 172–182